Man in the NFC

Build a NFC proxy tool from sketch

Haoqi Shan @ UnicornTeam
Agenda

- Who we are
- NFC & ISO14443A
- Competitions
- Yet another wheel?
- What is UniProxy?
- Master and Slave
- Issues in development
- Thanks, Q&A
Who we are

• **Unicorn team**
  • Internal security research team of Qihoo 360, founded in 2014
  • Focus on wireless/hardware hacking and defense
  • Security research division and hardware development division
  • Serial wireless researches published in Defcon/BlackHat
    • Low-cost GPS spoofing, Defcon 23
    • LTE redirection attack, Defcon 24
    • Attack on powerline communication, BlackHat USA 2016
    • ‘Ghost Telephonist’, Defcon 25/BlackHat USA 2017
  • Serial hacking tools developed
    • HackID/HackID Pro/SafeRFID/HackNFC, etc
    • https://unicorn.360.com
NFC & ISO14443A

• NFC
  • 13.56MHz
  • Low-cost
  • Not requires power
  • Well developed and deployed

• ISO14443A
  • Widely usage
  • Supports many applications
  • Security/Passport/BankCard
What we aim

- Credit card
  - QuickPass – Unipay (*)
- Starbucks POS machine
- XX: “I thought this question has been solved like a thousand times”
- More like a hacker
The way we used to hack

- Targeting protocols
  - Proxmark III (The Best Hardware way)
- Targeting data
  - NFCProxy
  - NFCGate
Why not?

- Proxmark III
  - Supports many protocols
  - Powerful
  - However, can’t hack credit card or we are all rich now

- NFCGate/NFCProxy
  - Based on Android
  - Modified firmware to relay NFC data
  - Monitor transmitted data
  - Rely on Wi-Fi
  - However, too much delay to complete whole payment procedure
Yet another wheel

- Inspired by mentioned brilliant hacking tool
- Faster (ms level)
- Lager ranger (50m, even more)
- Pure hardware solution (PN7462AU)
- Highly customization
- Completely self-designed and modify everything we need
What's UniProxy

- PN7462AU based NFC relay/proxy device
- Support ISO14443A protocol
- Targeting QuickPass(Unipay) credit cards
- Reader emulator, card emulator
- Point to Point wireless data transmission
- Easy to adapt to ISO 14443B/15693
Core of UniProxy

• Why PN7462AU?
  • NXP chip
  • 20 MHz Cortex-M0 core
  • Read/Write, Card Emulation & Peer-to-Peer Modes
  • Transmitter current up to 250 mA
  • Full MIFARE family support

• Architecture
  • Reader/Card Emulator
  • NRF24L01 wireless transmitter
  • Power supply
  • Antenna
Master (Back)

NFC Antenna

Lithium Battery
Process of Master (1)

Start → Init → No

RF-field

14443

→ Yes

Handshake

153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170

phOsal_Init();

/* Perform Platform Init */

status = phPlatform_Init(&sPlatform, bHalBufferTx,
CHECK_STATUS(status);
if(status != PH_ERR_SUCCESS) break;

/* Initialize Reader Library PAL/AL Components */

status = phApp_RdLibInit();
CHECK_STATUS(status);
if(status != PH_ERR_SUCCESS) break;

status = phpalI14443p4mC_SetConfig(
&spalI14443p4mC,
PHPAL_I14443P4MC_CONFIG_MODE,
RD_LIB_MODE_ISO);
if(status != PH_ERR_SUCCESS) break;
Process of Master (2)

- Communicate with card
- 14443A handshake and get parameters
  - Send parameters to card emulator
  - Receive response before timeout
    - Start block transmission
  - End

```c
/* Retrieve 14443-4A protocol parameter */
status = phpall14443p4a_GetProtocolParams(
pDataParams->pPal14443p4aDataParams,
&bCidEnabled,
&bCid,
&bNadSupported,
&bFwi,
&bFsd1,
&bFsci);
CHECK_STATUS(status);

/* Set 14443-4 protocol parameter */
status = phpall14443p4_SetProtocol(
pDataParams->pPal14443p4aDataParams,
PH_OFF,
bCid,
PH_OFF,
PH_OFF,
bFwi,
bFsd1,
bFsci);
CHECK_STATUS(status);
```
Process of Master (3)

1. Start block transmission
2. Wait response from card emulator before timeout
3. Forward data to real card, wait for real card response
   - Get response before timeout
     - Notify card emulator, communication is ended
     - I-Block data
5. Forward to card emulator
6. Process

Code snippet:
```c
void send(uint8_t *buff, uint16_t length)
{
    uint8_t count, i, length_last_packet;
    if (length <= MAX_SINGLE_PACKET_LENGTH)
    {
        #if 1
        packetbuff_send[0] = (uint8_t)length + 1;
        packetbuff_send[1] = CHAINING_NOT;
        memcpy(packetbuff_send + 2, buff, length);
        send_basic(packetbuff_send, length + 2);
        #endif
    }
```

Man in the NFC
Slave

NFC Antenna

Core

Power

→ 24 LoI

NFC Test U1.1

BoeTEAM
Process of Slave (1)

Start → init → Receive 14443

- No
- Yes

- Params from host
- Response success command
Process of Slave (2)

- Start interaction with reader emulator
- Init card emulator with received parameters
- Reader nearby
- Start interaction with received parameters
- Handshake with real reader
- Start block transmission

```c
switch(packetbuff_receive[1])
{
    case TYPE_BASIC:
        //uart_send(((uint8_t *)buffContext[0].pdwBuffAddr), buffContext[0].dwData);
        //debugPrint(((uint8_t *)buffContext[0].pdwBuffAddr), buffContext[0].dwData);
        //0x11 0x01 0x00  
        //uart_send(rsp_to_basicpara_rf, 5);//response to basicpara
        printf("type_basic\n");
        send(rsp_to_basicpara_rf,5);
        break;
    case TYPE_DATA:
        //phLED_SetStatus(LED_R);
        break;
}
```
Process of Slave (3)

- Start Block transmission
  - Received data
    - Yes
      - I-Block data
        - No
          - DESELECT command (S-Block)
            - Yes
              - Forward DESELECT to reader emulator and send DESELECT to real reader
            - No
              - Forward to real reader
    - No
      - Forward to reader emulator, send delay command after half waiting time
        - Yes
          - Received data from reader emulator
            - Yes
              - Forward to real reader
            - No
              - Flash error LED, self reset and send reset status to reader emulator
        - No
          - Finish
Issues in development

- First byte of UID
- Waiting/Wakeup time
- I/S/R – Block data
- ISO 14443A Part 4
- Power supply
- ...

Man in the NFC
Demo video
Summary

• What we learned
  • Read protocol document well
  • Better not developing without official support

• Further more
  • Improve transmission range up to 100 meters
  • Targeting security ID cards, HID iClass, Chinese ID
  • Self-compatibility
  • How?
References

- [NXP user guide](http://www.nxp.com/docs/en/user-guide/UM10883.pdf)
- [NFC Gate](https://github.com/nfcgate)
- [NFC Proxy](http://sourceforge.net/projects/nfcproxy)
- [ISO14443A](https://www.iso.org/standard/70172.html)
Thanks

- Hardware dev division of Unicorn Team, especially Jian Yuan, Chaoran Wang, and Yunding Jian
- Proxmark III
- NFCProxy
- NFCGate
Q&A

• Mail me: shanhaoqi@360.cn